

Fast Multiple-View L_2 Triangulation with Occlusion Handling

G. Chesi* and Y. S. Hung

Department of Electrical and Electronic Engineering

University of Hong Kong

Pokfulam Road, Hong Kong

Email: chesi@eee.hku.hk

Web: <http://www.eee.hku.hk/~chesi>

Abstract

Multiple-view L_2 triangulation is a key problem in computer vision. This paper addresses the standard case where all image points are available, and the case where some image points are not available. In the latter case, it is supposed that the unknown image point belongs to a known region such as a line segment or an ellipse, as it happens for instance due to occlusions. For this problem we propose two methods based on linear matrix inequalities (LMIs). The first method, named TFML, exploits the fundamental matrix and is fast (the average computational time with two and three views is 0.01 and 0.05 seconds on Matlab) at the expense of possible conservatism, which however it is shown to occur rarely in practice, and which can be immediately detected. The second method, named TPML, exploits the projection matrix, is slower, but allows one to reduce the conservatism by using techniques for optimization over polynomials. Various examples with synthetic and real data illustrate the proposed strategy.

Key words: Computer Vision; Multiple-View Triangulation; Occlusion; LMI.

1. Introduction

It is well-known that triangulation is an important problem in computer vision, as it allows one to estimate the 3D position of a point with respect to the camera locations. For instance, this is exploited in 3D object reconstruction, map estimation, robotic path-planning, etc. Triangulation can be

performed minimizing different norms of the reprojection error, and typically L_2 triangulation is preferred as it has been recognized to be more suitable for real applications.

There have been various contributions to this problem in the literature for L_2 triangulation. In [1, 2] the authors show how the exact solution of triangulation with two-views can be obtained by computing the roots of a one-variable polynomial of degree six. For triangulation with three-views, the exact solution is obtained in [3] by solving a system of polynomial equations through methods from computational commutative algebra, and in [4] through Groebner basis techniques. Multiple-view triangulations is considered for example in [5] via branch-and-bound algorithms. See also [6] that addresses the problem of verifying global minima and [7] that considers the case of points on lines. It is also useful mentioning some contributions for triangulation with different norms, such as [8, 9].

This paper addresses multiple-view L_2 triangulation in the standard case where all image points are available, and in the case where some image points are not available. In the latter case, it is supposed that the unknown image point belongs to a known region such as a line segment or an ellipse, as it happens for instance due to occlusions. For this problem we propose two methods based on linear matrix inequalities (LMIs). The first method, named TFML, is obtained through the fundamental matrix and is fast: the average computational time (ACT) in four well-known real sequences (dinosaur, corridor, house and college examples) for triangulations with two and three views are respectively 0.01 and 0.05 seconds on Matlab. This at the expense of possible conservatism, which however can be immediately detected, and which it is shown to occur rarely in practice. The second method, named TPML, is obtained through the projection matrix, is slower than the first method, but allows one to reduce the conservatism by using techniques based on sum of squares of polynomials (SOS) (see e.g. [10, 11]). The proposed strategy is illustrated through various examples with synthetic and real data.

2. Preliminaries

This section provides the notation adopted in the following sections and introduces the considered multi-view triangulation problems.

2.1. Notation

The notation adopted throughout the paper is as follows:

- \mathbf{M}^T : transpose of matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$;
- \mathbf{I}_n : $n \times n$ identity matrix;
- $\mathbf{0}_n$: $n \times 1$ null vector;
- \mathbf{e}_i : i -th column of \mathbf{I}_3 ;
- $SO(3)$: set of all 3×3 rotation matrices;
- $SE(3)$: $SO(3) \times \mathbb{R}^3$;
- $\|\mathbf{v}\|$: 2-norm of $\mathbf{v} \in \mathbb{R}^n$;
- $diag(\mathbf{M}_1, \mathbf{M}_2, \dots)$: block diagonal matrix having the square matrices $\mathbf{M}_1, \mathbf{M}_2, \dots$ on the main diagonal, and zero elsewhere.

Consider the situation where N cameras are observing a common point. Let $F_i = (\mathbf{R}_i, \mathbf{t}_i) \in SE(3)$ denote the coordinate frame of the i -th camera, where $\mathbf{R}_i \in SO(3)$ is the rotation matrix and $\mathbf{t}_i \in \mathbb{R}^3$ is the translation vector expressed with respect to a common reference coordinate frame $F^{ref} \in SE(3)$. The observed point in homogeneous coordinates are expressed as

$$\mathbf{X} = (x, y, z, 1)^T \quad (1)$$

where $x, y, z \in \mathbb{R}$ are expressed with respect to F^{ref} . The projection of \mathbf{X} onto the image plane of the i -th camera is given by

$$\mathbf{x}_i = (u_i, v_i, 1)^T \quad (2)$$

where $u_i, v_i \in \mathbb{R}$ are the screen coordinates along the horizontal and vertical directions. The relation between \mathbf{X} and \mathbf{x}_i is expressed by the projective law

$$d_i \mathbf{x}_i = \mathbf{P}_i \mathbf{X} \quad (3)$$

where $d_i \in \mathbb{R}$ and $\mathbf{P}_i \in \mathbb{R}^{3 \times 4}$ is the projection matrix given by

$$\mathbf{P}_i = \mathbf{K}_i [\mathbf{R}_i, \mathbf{t}_i] \quad (4)$$

with $\mathbf{K}_i \in \mathbb{R}^{3 \times 3}$ being the upper triangular matrix containing the intrinsic camera parameters of the i -th camera. The solution for \mathbf{x}_i in (3) is denoted by $\Phi(\mathbf{X}, \mathbf{P}_i)$ and has the expression

$$\Phi(\mathbf{X}, \mathbf{P}_i) = \frac{\mathbf{P}_i \mathbf{X}}{\mathbf{e}_3^T \mathbf{P}_i \mathbf{X}}. \quad (5)$$

2.2. Multiple-View L_2 Triangulation

Assume that the estimates $\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_N$ and $\hat{\mathbf{P}}_1, \dots, \hat{\mathbf{P}}_N$ of the image points $\mathbf{x}_1, \dots, \mathbf{x}_N$ and projection matrices $\mathbf{P}_1, \dots, \mathbf{P}_N$ are available. The standard triangulation problem is compute an estimate $\hat{\mathbf{X}}$ of \mathbf{X} from $\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_N$ and $\hat{\mathbf{P}}_1, \dots, \hat{\mathbf{P}}_N$. Typically the estimate $\hat{\mathbf{X}}$ is defined as the minimizer of the mean square re-projection error in L_2 norm, i.e.

$$\mathbf{X}^* = \arg \min_{\mathbf{X}} \sum_{i=1}^N \|\Phi(\mathbf{X}, \hat{\mathbf{P}}_i) - \hat{\mathbf{x}}_i\|^2. \quad (6)$$

In this paper we address a more general triangulation problem as follows:

$$\mathbf{X}^* = \arg \min_{\mathbf{X}} \sum_{i=1}^N d(\Phi(\mathbf{X}, \hat{\mathbf{P}}_i), \mathcal{A}_i)^2 \quad (7)$$

where \mathcal{A}_i is the region of admissible values for $\hat{\mathbf{x}}_i$, and $d(\Phi(\mathbf{X}, \hat{\mathbf{P}}_i), \mathcal{A}_i)$ is the euclidean distance from the projection $\Phi(\mathbf{X}, \hat{\mathbf{P}}_i)$ to the region \mathcal{A}_i , i.e.

$$d(\Phi(\mathbf{X}, \hat{\mathbf{P}}_i), \mathcal{A}_i) = \inf_{\mathbf{x} \in \mathcal{A}_i} \|\Phi(\mathbf{X}, \hat{\mathbf{P}}_i) - \mathbf{x}\|. \quad (8)$$

This allows one to consider the case where the measurements $\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_N$ are not exactly known (for instance, due to occlusions), and can vary inside the regions $\mathcal{A}_1, \dots, \mathcal{A}_N$. Some cases of interests are

$$\mathcal{A}_i = \{\hat{\mathbf{x}}_i\} \quad (9)$$

i.e. the measurement is known to be $\hat{\mathbf{x}}_i$,

$$\mathcal{A}_i = \left\{ \hat{\mathbf{x}}_i + y_i \hat{\mathbf{l}}_i, \quad y_i \in [y_i^-, y_i^+] \right\} \quad (10)$$

i.e. the measurement is known to lie on the line segment with extremes $\hat{\mathbf{x}}_i + y_i^- \hat{\mathbf{l}}_i$ and $\hat{\mathbf{x}}_i + y_i^+ \hat{\mathbf{l}}_i$,

$$\mathcal{A}_i = \left\{ \mathbf{x} : (\mathbf{x} - \hat{\mathbf{x}}_i)^T \text{diag}(\mathbf{E}_i^{-1}, 0)(\mathbf{x} - \hat{\mathbf{x}}_i) = 1 \right\} \quad (11)$$

i.e. the measurement is known to lie on the border of an ellipse centered at $\hat{\mathbf{x}}_i$ and described by the positive definite matrix $\mathbf{E}_i \in \mathbb{R}^{2 \times 2}$, and

$$\mathcal{A}_i = \left\{ \mathbf{x} : (\mathbf{x} - \hat{\mathbf{x}}_i)^T \text{diag}(\mathbf{E}_i^{-1}, 0)(\mathbf{x} - \hat{\mathbf{x}}_i) \leq 1 \right\} \quad (12)$$

i.e. the measurement is known to lie inside such an ellipse.

It is worth observing that:

- (9) is the typical case when the image point is available;
- (10)–(11) are cases when the image point is known to belong to the border of an object. In particular, (10) considers line segments (e.g., the edges of books, boxes, screens, rooms, etc) while (11) considers ellipses (e.g., the borders of cups, road signals, wheels, etc);
- (12) is the case when the image point is known to lie inside an ellipse.

Before proceeding it is worth observing that, in general, the regions $\mathcal{A}_1, \dots, \mathcal{A}_N$ should be chosen as small as possible compatibly with the information available in the views. In fact, by enlarging these regions beyond a certain value, one would consider image points that are far away from the real projections, i.e. image points that are obviously “wrong” for the triangulation problem.

3. Proposed Strategy

Let us define the optimal cost of the generalized triangulation problem as

$$\mu^* = \sum_{i=1}^N d(\Phi(\mathbf{X}^*, \hat{\mathbf{P}}_i), \mathcal{A}_i)^2 \quad (13)$$

where \mathbf{X}^* is the solution of (7). The first step consists of rewriting μ^* by using variables in the image domain rather than in the 3D space. We hence obtain

$$\mu^* = \min_{\mathbf{x}_1, \dots, \mathbf{x}_N} \sum_{i=1}^N d(\mathbf{x}_i, \mathcal{A}_i)^2 \quad \text{s.t.} \quad (\mathbf{x}_1, \dots, \mathbf{x}_N) \in \hat{\mathcal{C}} \quad (14)$$

where $\hat{\mathcal{C}}$ is the set of admissible values for the variables $\mathbf{x}_1, \dots, \mathbf{x}_N$, which is given by

$$\hat{\mathcal{C}} = \left\{ \left(\Phi(\mathbf{X}, \hat{\mathbf{P}}_1), \dots, \Phi(\mathbf{X}, \hat{\mathbf{P}}_N) \right) \text{ for some } \mathbf{X} \right\}. \quad (15)$$

Then, we eliminate the region \mathcal{A}_i from the cost function as follows:

1. for those i for which \mathcal{A}_i is a point we explicitly write $d(\mathbf{x}_i, \mathcal{A}_i)^2$ as

$$\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2; \quad (16)$$

2. we introduce an additional variable $y_i \in \mathbb{R}$ for each of those i for which \mathcal{A}_i is a line segment as in (10). The term $d(\mathbf{x}_i, \mathcal{A}_i)^2$ is

$$\|\mathbf{x}_i - \hat{\mathbf{x}}_i - y_i \hat{\mathbf{l}}_i\|^2; \quad (17)$$

3. we introduce an additional variable $\mathbf{z}_i = (\tilde{u}_i, \tilde{v}_i, 1)^T$ with $\tilde{u}_i, \tilde{v}_i \in \mathbb{R}$ for each of the remaining values of i . The term $d(\mathbf{x}_i, \mathcal{A}_i)^2$ is

$$\|\mathbf{x}_i - \mathbf{z}_i\|^2. \quad (18)$$

Let us gather all scalar unknowns of the variables $\mathbf{x}_1, \dots, \mathbf{x}_N, \dots, y_i, \dots, \mathbf{z}_i, \dots$ into the vector

$$\mathbf{w} = (u_1, v_1, \dots, u_N, v_N, \dots, y_i, \dots, \tilde{u}_i, \tilde{v}_i, \dots, 1)^T. \quad (19)$$

Since (16)–(18) are quadratic functions in \mathbf{w} , it follows that (14) is equivalent to the new problem

$$\mu^* = \min_{\mathbf{w}} \mathbf{w}^T \mathbf{C} \mathbf{w} \quad \text{s.t.} \quad \begin{cases} (\mathbf{x}_1, \dots, \mathbf{x}_N) \in \hat{\mathcal{C}} \\ g_i(\mathbf{w}) \geq 0 \quad \forall i = 1, \dots, N_1 \\ h_i(\mathbf{w}) = 0 \quad \forall i = 1, \dots, N_2 \end{cases} \quad (20)$$

where \mathbf{C} is a positive semidefinite matrix, $g_1(\mathbf{w}), \dots, g_{N_1}(\mathbf{w})$ describe inequality constraints on \mathbf{w} (e.g., deriving from (10) or (12)), and $h_1(\mathbf{w}), \dots, h_{N_2}(\mathbf{w})$ describe equality constraints on \mathbf{w} (e.g., deriving from (11)). At this point we propose two possible strategies for solving the problem (20).

3.1. TFML Method

Here we describe a first method for triangulation based on fundamental matrices and LMIs, that we refer to as TFML method. For this method we suppose that the functions $g_i(\mathbf{w})$ and $h_i(\mathbf{w})$ in (20) are quadratic.

In order to solve (20), we introduce the following modified problem:

$$\mu_F^* = \min_{\mathbf{w}} \mathbf{w}^T \mathbf{C} \mathbf{w} \quad \text{s.t.} \quad \begin{cases} (\mathbf{x}_1, \dots, \mathbf{x}_N) \in \hat{\mathcal{C}}_F \\ g_i(\mathbf{w}) \geq 0 \quad \forall i = 1, \dots, N_1 \\ h_i(\mathbf{w}) = 0 \quad \forall i = 1, \dots, N_2 \end{cases} \quad (21)$$

where the set $\hat{\mathcal{C}}_F$ is defined as

$$\hat{\mathcal{C}}_F = \left\{ (\mathbf{x}_1, \dots, \mathbf{x}_N) : \mathbf{x}_i^T \hat{\mathbf{F}}_{i,j} \mathbf{x}_j = 0 \right. \\ \left. \forall i = 1, \dots, N-1, j = i+1, \dots, N \right\} \quad (22)$$

and $\hat{\mathbf{F}}_{i,j} \in \mathbb{R}^{3 \times 3}$ is the estimate of the fundamental matrix between the i -th and the j -th view (this estimate can be simply computed from the estimates of the projection matrices $\hat{\mathbf{P}}_i$ and $\hat{\mathbf{P}}_j$ as described for example in [2]).

The problem (21) is a relaxation of the original problem (20). In fact, one has

$$\hat{\mathcal{C}} \subseteq \hat{\mathcal{C}}_F \quad (23)$$

which hence implies

$$\mu_F^* \leq \mu^*. \quad (24)$$

Therefore, the solution of the modified problem (21) may be different from the solution of (20) since the variables are allowed to vary in a possibly larger set. Nevertheless, we will verify in Section 4 that such a case seldom occurs in practice.

Let us describe now the proposed strategy for solving (21). Problem (21) can be rewritten as

$$\mu_F^* = \min_{\mathbf{w}} \mathbf{w}^T \mathbf{C} \mathbf{w} \quad \text{s.t.} \quad \begin{cases} g_i(\mathbf{w}) \geq 0 \quad \forall i = 1, \dots, N_1 \\ h_i(\mathbf{w}) = 0 \quad \forall i = 1, \dots, N_3 \end{cases} \quad (25)$$

where the constraints $\mathbf{x}_i^T \hat{\mathbf{F}}_{i,j} \mathbf{x}_j = 0$ defining the set $\hat{\mathcal{C}}_F$ are included via additional equalities $h_i(\mathbf{w}) = 0$. Let $s_0, s_1^G, \dots, s_{N_1}^G, s_1^H, \dots, s_{N_3}^H$ be scalar variables, and let us define

$$q(\mathbf{w}) = \mathbf{w}^T \mathbf{C} \mathbf{w} - s_0 - \sum_{i=1}^{N_1} s_i^G g_i(\mathbf{w}) - \sum_{i=1}^{N_3} s_i^H h_i(\mathbf{w}) \quad (26)$$

where \mathbf{s} is the vector

$$\mathbf{s} = (s_0, s_1^G, \dots, s_{N_1}^G, s_1^H, \dots, s_{N_3}^H)^T. \quad (27)$$

Let us define

$$\mu_1 = \sup_{\mathbf{s}} s_0 \quad \text{s.t.} \quad \begin{cases} q(\mathbf{w}) \geq 0 \quad \forall \mathbf{w} \\ s_i^G \geq 0, \quad i = 1, \dots, N_1. \end{cases} \quad (28)$$

It follows that

$$\mu_1 \leq \mu_F^*. \quad (29)$$

In fact, since $q(\mathbf{w}) \geq 0$ for all \mathbf{w} one has

$$\mathbf{w}^T \mathbf{C} \mathbf{w} \geq s_0 + \sum_{i=1}^{N_1} s_i^G g_i(\mathbf{w}) + \sum_{i=1}^{N_3} s_i^H h_i(\mathbf{w})$$

and since $s_i^G \geq 0$ it follows that, for any \mathbf{w} such that the constraints in (25) hold, one has

$$\begin{aligned} s_i^G g_i(\mathbf{w}) &\geq 0 \\ s_i^H h_i(\mathbf{w}) &= 0 \end{aligned}$$

and hence

$$\mathbf{w}^T \mathbf{C} \mathbf{w} \geq s_0$$

which implies that s_0 is a lower bound of μ_F^* whenever the constraints in (28) hold. This relaxation procedure exploited in (28) is known as S-procedure, see e.g. [11].

It turns out that (28) is that it is a convex optimization. Indeed, since $q(\mathbf{w})$ is quadratic, (28) can be equivalently rewritten as

$$\mu_1 = \sup_{\mathbf{s}} s_0 \quad \text{s.t.} \quad \begin{cases} \mathbf{Q} \geq 0 \\ s_i^G > 0, \quad i = 1, \dots, N_1 \end{cases} \quad (30)$$

where \mathbf{Q} is the symmetric matrix satisfying

$$q(\mathbf{w}) = \mathbf{w}^T \mathbf{Q} \mathbf{w}.$$

The optimization problem (30) belongs to the class of eigenvalue problems (EVPs), which are convex optimizations with LMI constraints. Specifically, an EVP is the minimization of a linear cost function over a convex feasibility set described by a set of LMIs. EVPs can be solved in various ways, for instance through ellipsoidal algorithms or through interior-point methods. See e.g. [12] about EVPs.

Once (30) is solved, one builds a candidate solution for the original problem (7) as follows. Let $\bar{\mathbf{w}}$ be a vector with form as in (19) satisfying

$$\bar{\mathbf{Q}} \bar{\mathbf{w}} = \lambda_{\min}(\bar{\mathbf{Q}}) \bar{\mathbf{w}} \quad (31)$$

where $\bar{\mathbf{Q}}$ is the matrix \mathbf{Q} evaluated at the optimum of (30). Let us extract $\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_N$ from $\bar{\mathbf{w}}$. The candidate solution of the generalized triangulation problem (7) is computed as

$$\hat{\mathbf{X}} = \min_{\mathbf{X}} \sum_{i=1}^N \left\| \hat{\mathbf{P}}_i \mathbf{X} - \bar{\mathbf{x}}_i \mathbf{e}_3^T \hat{\mathbf{P}}_i \mathbf{X} \right\|^2 \quad (32)$$

which is a linear least-squares minimization.

In general, there is no guarantee that this candidate lies in front of the cameras. One can ensure this property by introducing the linear constraints $\mathbf{e}_3^T \hat{\mathbf{P}}_i \mathbf{X} > 0$ in (32) (observe that the new least-square problem is still convex and can be readily solved). It is worth mentioning, however, that these constraints are typically superfluous, see for instance the examples in Section 4 where $\hat{\mathbf{X}}$ always lies in front of the cameras though these constraints are not imposed.

What can be said about the optimality of the found solution? Let us define

$$\mu_2 = \sum_{i=1}^N d(\Phi(\hat{\mathbf{X}}, \hat{\mathbf{P}}_i), \mathcal{A}_i)^2. \quad (33)$$

Clearly, μ_2 is an upper bound of the solution of the original problem (7) since it is the cost function evaluated in a feasible point. Hence, at this point one has a lower and an upper bound of μ^* :

$$\mu^* \in [\mu_1, \mu_2]. \quad (34)$$

From this, it is clearly possible to derive an immediate test for establishing whether $\hat{\mathbf{X}}$ is the optimal solution of (7). Indeed:

$$\begin{aligned} \mu_1 &= \mu_2 \\ \Downarrow \\ \mu^* &= \mu_1 \text{ and } \hat{\mathbf{X}} \text{ is optimal (i.e., solution of (7)).} \end{aligned} \quad (35)$$

The TFML method described in this section exploits the fundamental matrices among the available views, and hence its numerical solution can be sensitive to short baselines. Also, $\hat{\mathbf{X}}$ can be not optimal because the set of admissible image points $\hat{\mathcal{C}}_F$ in (21) can be larger than the set of admissible image points $\hat{\mathcal{C}}$ in (20) according to (23). Nevertheless, the cases where $\hat{\mathbf{X}}$ is not optimal seem to be rare in practice, see the various examples in Section 4. Moreover, when $\hat{\mathbf{X}}$ is not optimal, the proposed strategy provides a lower and an upper bound of the solution of the original problem (7) as expressed by (34).

3.2. TPML Method

Here we describe another method for triangulation based on projection matrices and LMIs, that we refer to as TPML method. For this method we suppose that the functions $g_i(\mathbf{w})$ and $h_i(\mathbf{w})$ in (20) are polynomial.

In order to solve (20), let us start by observing that the set $\hat{\mathcal{C}}$ in (15) can be rewritten as

$$\hat{\mathcal{C}} = \left\{ (\mathbf{x}_1, \dots, \mathbf{x}_N) : \mathbf{x}_i = \frac{\hat{\mathbf{P}}_i \mathbf{X}}{\mathbf{e}_3^T \hat{\mathbf{P}}_i \mathbf{X}} \quad \forall i = 1, \dots, N \text{ for some } \mathbf{X} \right\}$$

and hence

$$\hat{\mathcal{C}} = \left\{ (\mathbf{x}_1, \dots, \mathbf{x}_N) : \hat{\mathbf{P}}_i \mathbf{X} = \mathbf{x}_i \mathbf{e}_3^T \hat{\mathbf{P}}_i \mathbf{X} \quad \forall i = 1, \dots, N \text{ for some } \mathbf{X} \right\}.$$

We have that $\hat{\mathbf{P}}_i \mathbf{X} = \mathbf{x}_i \mathbf{e}_3^T \hat{\mathbf{P}}_i \mathbf{X}$ for all $i = 1, \dots, N$ if and only if

$$\mathbf{M}(\mathbf{w}) \mathbf{X} = \mathbf{0}_{2N} \quad (36)$$

where

$$\mathbf{M}(\mathbf{w}) = \begin{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \hat{\mathbf{P}}_1 - (u_1, v_1)^T \mathbf{e}_3^T \hat{\mathbf{P}}_1 \\ \vdots \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \hat{\mathbf{P}}_N - (u_N, v_N)^T \mathbf{e}_3^T \hat{\mathbf{P}}_N \end{pmatrix}.$$

As next step, let us observe that (36) admits a solution \mathbf{X} if and only if

$$\tilde{h}_i(\mathbf{w}) = 0 \quad \forall i = 1, 2, \dots$$

where $\tilde{h}_i(\mathbf{w})$ are polynomials representing the 4×4 determinants of $\mathbf{M}(\mathbf{w})$, since this implies that $\mathbf{M}(\mathbf{w})$ does not have full rank. Hence, $\hat{\mathcal{C}}$ can be rewritten as

$$\hat{\mathcal{C}} = \left\{ (\mathbf{x}_1, \dots, \mathbf{x}_N) : \tilde{h}_i(\mathbf{w}) = 0, \quad i = 1, 2, \dots \right\}.$$

Therefore, (20) can be rewritten as

$$\mu^* = \min_{\mathbf{w}} \mathbf{w}^T \mathbf{C} \mathbf{w} \quad \text{s.t.} \quad \begin{cases} g_i(\mathbf{w}) \geq 0 \quad \forall i = 1, \dots, N_1 \\ h_i(\mathbf{w}) = 0 \quad \forall i = 1, \dots, N_4 \end{cases} \quad (37)$$

where the equalities $\tilde{h}_i(\mathbf{w}) = 0$ have been gathered with those already existing in (20).

Problem (37) can be solved via LMIs by using the Positivstellensatz [13], which is a natural extension of the S-procedure used in Section 3.1. Let us define

$$p(\mathbf{w}) = \mathbf{w}^T \mathbf{C} \mathbf{w} - t_0 - \sum_{i=1}^{N_1} t_i^I(\mathbf{w}) g_i(\mathbf{w}) - \sum_{i=1}^{N_4} t_i^E(\mathbf{w}) h_i(\mathbf{w})$$

where t_0 is a scalar variable and $t_i^I(\mathbf{w}), t_i^E(\mathbf{w})$ are polynomial variables, and

$$\mu_3 = \sup_{t_0, t_i^I(\mathbf{w}), t_i^E(\mathbf{w})} t_0 \quad \text{s.t.} \quad \begin{cases} p(\mathbf{w}) \text{ is SOS} \\ t_i^I(\mathbf{w}) \text{ is SOS, } i = 1, \dots, N_1 \end{cases} \quad (38)$$

where SOS means to be sum of squares of polynomials. Analogous to (29), it is easy to verify that

$$\mu_3 \leq \mu_F^*. \quad (39)$$

The interest for the optimization problem (38) is that each condition to be SOS is equivalent to an LMI by using the square matrix representation (SMR) [10]. Hence, the optimization (38) is an EVP similarly to (28). Moreover, the conservatism of this method can be reduced by increasing the degree of the auxiliary polynomials $t_i^I(\mathbf{w}), t_i^E(\mathbf{w})$ [11]. See also [14, 15] for other applications of the SMR in computer vision.

Once (38) is solved, one builds a candidate solution $\hat{\mathbf{X}}$ for the original problem in a way similar to (31)–(32). Specifically, let \mathbf{S} be the SMR matrix used to assess the SOS property in (38), i.e. such that \mathbf{S} is positive semidefinite and

$$p(\mathbf{w}) = \mathbf{b}(\mathbf{w})^T \mathbf{S} \mathbf{b}(\mathbf{w}) \quad (40)$$

where $\mathbf{b}(\mathbf{w})$ is a vector containing monomials in \mathbf{w} , e.g. $\mathbf{b}(\mathbf{w}) = (\mathbf{w}^T, u_1^2, u_1 v_1, v_1^2, \dots)^T$. Let \mathbf{c} be the eigenvector corresponding to the minimum eigenvalue of \mathbf{S} , and define $\bar{\mathbf{w}}$ as the subpart of \mathbf{c} corresponding to \mathbf{w} in $\mathbf{b}(\mathbf{w})$. Then, one takes the image points $\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_N$ from $\bar{\mathbf{w}}$, and estimates $\hat{\mathbf{X}}$ as in (32).

From the so computed candidate $\hat{\mathbf{X}}$, we define the quantity

$$\mu_4 = \sum_{i=1}^N d(\Phi(\hat{\mathbf{X}}, \hat{\mathbf{P}}_i), \mathcal{A}_i)^2. \quad (41)$$

Similarly to (34)–(35), one has

$$\mu^* \in [\mu_3, \mu_4] \quad (42)$$

and

$$\begin{aligned} \mu_3 &= \mu_4 \\ \Downarrow \\ \mu^* &= \mu_3 \text{ and } \hat{\mathbf{X}} \text{ is optimal (i.e., solution of (7)).} \end{aligned} \tag{43}$$

4. Results with Synthetic and Real Data

In this section we present some examples of the proposed strategy. For ease of evaluation, we denote with μ^* , μ_1 , etc, their normalized value $\sqrt{\mu^*/(2N)}$, $\sqrt{\mu_1/(2N)}$, etc.

4.1. Illustrative Cases

Here we present some illustrative examples with the projections matrices

$$\begin{aligned} \hat{\mathbf{P}}_1 &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}, & \hat{\mathbf{P}}_2 &= \begin{pmatrix} -1 & -1 & -1 & 0 \\ 1 & 0 & -1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \\ \hat{\mathbf{P}}_3 &= \begin{pmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ -1 & -1 & 0 & 1 \end{pmatrix}, & \hat{\mathbf{P}}_4 &= \begin{pmatrix} 0 & -1 & -1 & 0 \\ 0 & 1 & -1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}. \end{aligned} \tag{44}$$

Examples SA2–SA4. The first problem we consider is the standard triangulation problem with $\hat{\mathbf{x}}_i = (0, 0, 1)^T$ for all $i = 1, \dots, 4$. We denote by Example SA k this problem based on the first k views, for $k = 2, 3, 4$.

Examples SB2–SB4. As the second problem, we consider that some projections $\hat{\mathbf{x}}_i$ are unknown (e.g., due to occlusions), while the others are $\hat{\mathbf{x}}_i = (0, 0, 1)^T$. Specifically:

1. (Example SB2) Two views are available, and $\hat{\mathbf{x}}_1$ lies on a disc:

$$\begin{aligned} \min_{\mathbf{X}} d(\Phi(\mathbf{X}, \hat{\mathbf{P}}_1), \mathcal{A}_1)^2 + \|\Phi(\mathbf{X}, \hat{\mathbf{P}}_2) - \hat{\mathbf{x}}_2\|^2 \\ \mathcal{A}_1 = \{\mathbf{x} : \|\mathbf{x} - (0, 0, 1)^T\|^2 \leq 0.1^2\}. \end{aligned}$$

2. (Example SB3) Three views are available, and $\hat{\mathbf{x}}_1$ lies on a line segment:

$$\begin{aligned} \min_{\mathbf{X}} d(\Phi(\mathbf{X}, \hat{\mathbf{P}}_1), \mathcal{A}_1)^2 + \sum_{i=2}^3 \|\Phi(\mathbf{X}, \hat{\mathbf{P}}_2) - \hat{\mathbf{x}}_i\|^2 \\ \mathcal{A}_1 = \{(0, 0, 1)^T + y_1(1, 1, 0)^T, y_1 \in [-0.1, 0.1]\} \end{aligned}$$

3. (Example SB4) Four views are available, $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ lie on ellipses, and $\hat{\mathbf{x}}_3$ lies on a line segment:

$$\begin{aligned} \min_{\mathbf{X}} \sum_{i=1}^3 d(\Phi(\mathbf{X}, \hat{\mathbf{P}}_i), \mathcal{A}_i)^2 + \|\Phi(\mathbf{X}, \hat{\mathbf{P}}_4) - \hat{\mathbf{x}}_4\|^2 \\ \mathcal{A}_1 = \{\mathbf{x} : 2u_1^2 + v_1^2 \leq 0.1^2\} \\ \mathcal{A}_2 = \{\mathbf{x} : u_2^2 + u_2v_2 + 2v_2^2 \leq 0.1^2\} \\ \mathcal{A}_3 = \{(0, 0, 1)^T + y_3(1, 1, 0)^T, y_3 \in [-0.1, 0.1]\} \end{aligned}$$

Examples SC2–SC4. As the third problem, we consider that all projections $\hat{\mathbf{x}}_i$ are unknown (as it is the real case since image noise is always present), in particular that $\hat{\mathbf{x}}_i$ lies on the disk

$$\mathcal{A}_i = \{\mathbf{x} : u_i^2 + v_i^2 \leq 0.1^2\}$$

for all $i = 1, \dots, 4$. We denote with Example SC k this problem by using the first k views, for $k = 2, 3, 4$.

Table 1 shows the results obtained for Examples SA, SB and SC by using the TFML method. We can observe that in all these examples $\mu_1 = \mu_2$, i.e. the solution is optimal according to (35), and hence $\mu^* = \mu_1 = \mu_2$. Table 1 also shows the numerical complexity in terms of number of scalar variables in (30). Figures 1–3 illustrate the solutions of Examples SB2–SB4.

4.2. A Conservative Case

Here we want to show a case where the TFML method yields a conservative result. Let us consider the triangulation problem (6) with three views by using the projection matrices $\hat{\mathbf{P}}_1$, $\hat{\mathbf{P}}_2$ and $\hat{\mathbf{P}}_3$ in (44). Let us select

$$\hat{\mathbf{x}}_1 = (0.9, -0.9, 1)^T, \quad \hat{\mathbf{x}}_2 = (0.6, 2, 1)^T, \quad \hat{\mathbf{x}}_3 = (2, 1.3, 1)^T.$$

We obtain the following results:

$$\mu_1 = 0.384, \quad \mu_2 = 0.455.$$

As we can see, the lower bound μ_1 and the upper bound μ_2 are different in this case.

As explained at the end of Section 3.1, this happens because the set of admissible image points $\hat{\mathcal{C}}_F$ in (21) can be larger than the set of admissible

Example	μ_1 [pixel]	μ_2 [pixel]	$\hat{\mathbf{X}}$	N_v
SA2	0.118	0.118	$(-0.273, -0.182, 0.636)^T$	2
SA3	0.132	0.132	$(-0.303, -0.161, 0.799)^T$	4
SA4	0.162	0.162	$(-0.232, -0.335, 0.697)^T$	7
SB2	0.075	0.075	$(-0.310, -0.207, 0.632)^T$	3
SB3	0.107	0.107	$(-0.349, -0.208, 0.784)^T$	5
SB4	0.110	0.110	$(-0.160, -0.364, 0.663)^T$	10
SC2	0.049	0.049	$(-0.250, -0.167, 0.639)^T$	4
SC3	0.062	0.062	$(-0.301, -0.164, 0.793)^T$	7
SC4	0.096	0.096	$(-0.187, -0.319, 0.718)^T$	11

Table 1: Results for the examples SA, SB and SC: lower bound μ_1 , upper bound μ_2 , candidate solution $\hat{\mathbf{X}}$, and total number of scalar variables N_v in (30). Since $\mu_1 = \mu_2$, one has that $\mu^* = \mu_1$, and $\hat{\mathbf{X}}$ is the solution of the triangulation problem.

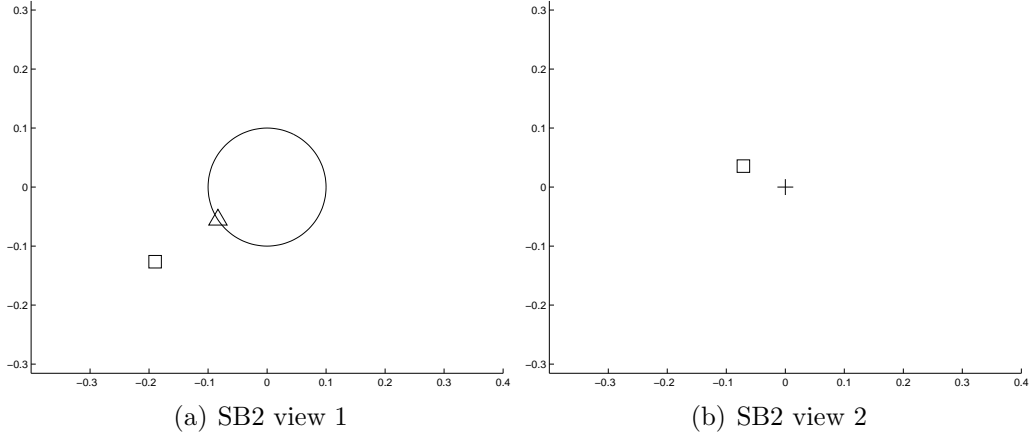


Figure 1: Example SB2: problem data (disc in figure a and cross in figure b) and projections of $\hat{\mathbf{X}}$ (squares). The triangle is the closest point of the disc to the projection of $\hat{\mathbf{X}}$.

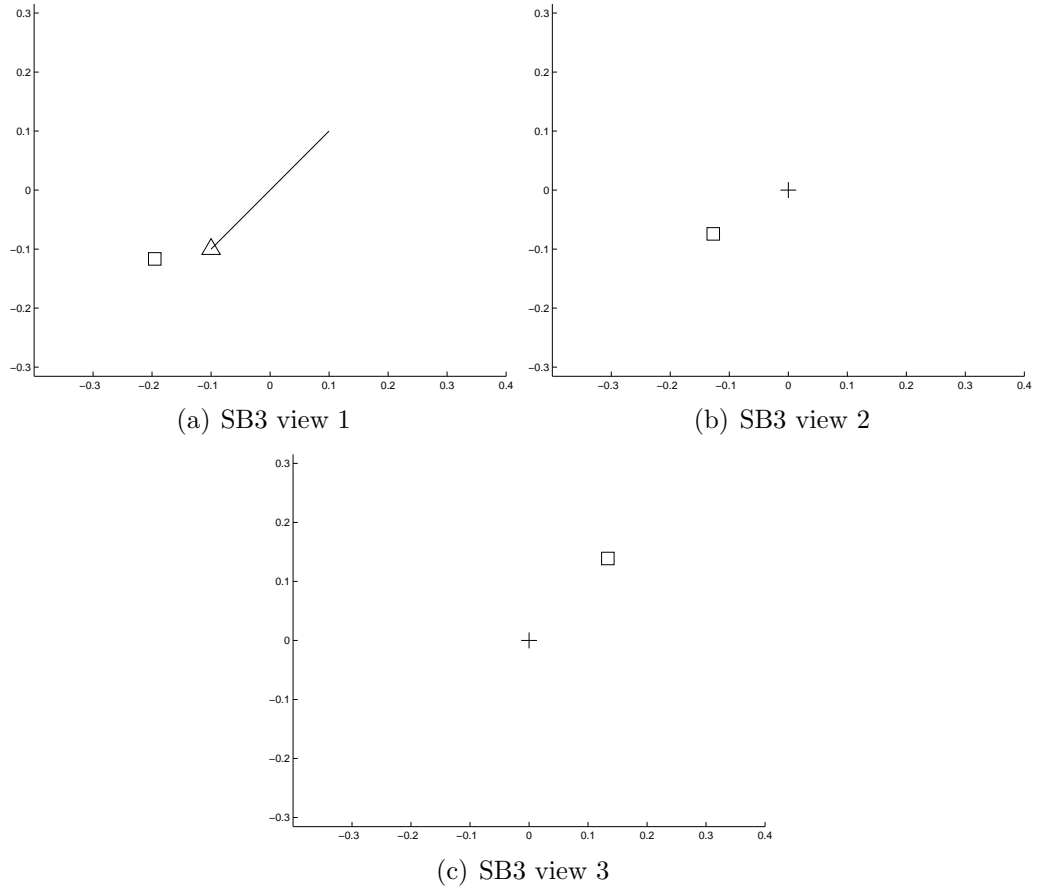


Figure 2: Example SB3: problem data (line segment in figure a and crosses in figures b and c) and projections of $\hat{\mathbf{X}}$ (squares). The triangle is the closest point of the line segment to the projection of $\hat{\mathbf{X}}$.

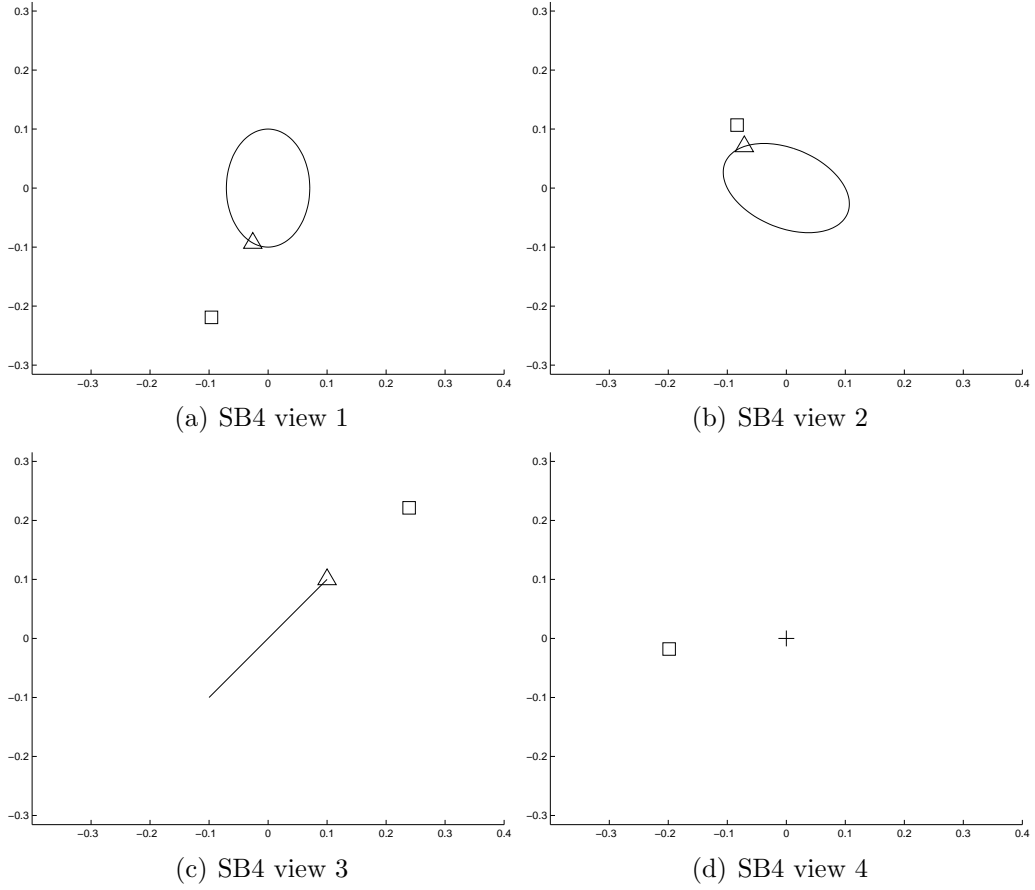


Figure 3: Example SB4: problem data (ellipses in figures b and c, line segment in figure c, and cross in figure d) and projections of $\hat{\mathbf{X}}$ (squares). The triangles are the closest points on their respective regions of uncertainty to the projections of $\hat{\mathbf{X}}$.

image points $\hat{\mathcal{C}}$ in (20). Indeed, the quantities $\bar{\mathbf{x}}_1$, $\bar{\mathbf{x}}_2$ and $\bar{\mathbf{x}}_3$ used to build $\hat{\mathbf{X}}$ in (32) are

$$\bar{\mathbf{x}}_1 = (1.039, -0.808, 1)^T, \quad \bar{\mathbf{x}}_2 = (-0.080, 2.340, 1)^T, \quad \bar{\mathbf{x}}_3 = (1.708, 0.862, 1)^T$$

and it turns out that $(\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \bar{\mathbf{x}}_3)$ belongs to $\hat{\mathcal{C}}_F$ but does not belong to $\hat{\mathcal{C}}$.

We hence use the TPML method. By selecting $p(\mathbf{w})$ of degree 4, which is the smallest degree for this polynomial, there are 301 variables in (38). The found solution is

$$\mu_3 = 0.452, \quad \mu_4 = 0.452, \quad \hat{\mathbf{X}} = (1.424, -1.238, 0.116)^T.$$

As we can see, μ_3 and μ_4 are equal, i.e. the optimal solution has been found and $\mu^* = \mu_3 = \mu_4 = 0.452$.

4.3. Statistics with Large Number of Views

Here we want to show the results obtained with random configurations using large number of views. The screen size of each camera is 800×800 pixels, and all the intrinsic camera matrices are chosen as

$$\mathbf{K}_i = \begin{pmatrix} 800 & 0 & 400 \\ 0 & 800 & 400 \\ 0 & 0 & 1 \end{pmatrix}.$$

Each random configuration is generated as follows:

1. a point is randomly located (with uniform distribution) inside the trapezoid shown in Figure 4, which is the cone of the field of view of the first camera delimited at two different depths;
2. the other cameras are randomly located (with uniform distribution) inside the box shown in Figure 4 under the constraint that the the previously generated point lies in their field of view;
3. the estimates of the projection matrices $\hat{\mathbf{P}}_i$ are obtained by multiplying each entry of \mathbf{K}_i , \mathbf{t}_i and of the vector of exponential coordinates of \mathbf{R}_i times a random variable with uniform distribution in the interval $[0.995, 1.005]$ (i.e., with size equal to 1%);
4. the estimates of the image projections $\hat{\mathbf{x}}_i$ are obtained by adding to each component of \mathbf{x}_i a random variable (with uniform distribution) in the interval $[-\eta, \eta]$ pixels, where η indicates the level of the noise and will be specified in the sequel.

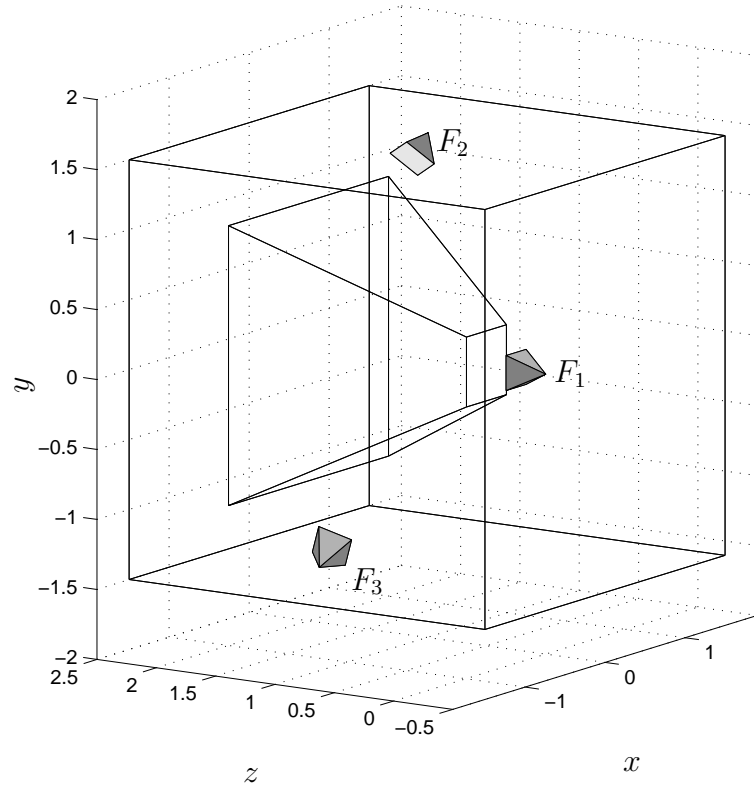


Figure 4: A point is randomly located inside the trapezoid, and N cameras are randomly located under the constraint that the point must lie in their field of view.

We have generated random configurations as described above with three different number of views ($N = 10, 30, 50$), three different values of the noise ($\eta = 1, 2, 3$ pixels) and three different types of data, in particular:

1. Type “P”: a point per view;
2. Type “L”: a point per view in $N/2$ views, and a line segment of length 2η pixels per view in the other $N/2$ views;
3. Type “E”: a point per view in $N/2$ views, and a disc of diameter 2η pixels per view in the other $N/2$ views.

The line segments and the ellipses are centered in $\hat{\mathbf{x}}_i$, and the orientation of the line segments is randomly chosen. For each combination of number of views, level of noise, and type of data there are 400 configurations, and hence the total number of configurations is $3 \times 3 \times 3 \times 400 = 10800$.

Table 2 shows the obtained results. The quantity ξ is the relative error between μ_1 and μ_2 , i.e.

$$\xi = \frac{\mu_2 - \mu_1}{\mu_2}. \quad (45)$$

As we can see, the occurrences of cases with $\mu_1 \neq \mu_2$ are indeed rare¹, moreover the maximum relative error is quite small in such cases.

4.4. A Case with Multiple Solutions

In this example we want to show the behavior of the proposed strategy in the presence of multiple solutions. Let us consider a case with two views ($N = 2$) where $\hat{\mathbf{P}}_1$ and $\hat{\mathbf{P}}_2$ are as in (44). We suppose that the information for the second view is the image point $\hat{\mathbf{x}}_2 = (0, 0, 1)^T$, while for the first view is the line segment

$$\mathcal{A}_1 = \{(0, -0.5, 1)^T + y_1(1, -1.5, 0)^T, y_1 \in [-1, 1]\}.$$

Since the fundamental matrix is given by

$$\hat{\mathbf{F}}_{1,2} = \begin{pmatrix} 0.000 & 0.000 & 0.775 \\ 0.000 & 0.000 & 0.516 \\ 0.516 & -0.258 & 0.258 \end{pmatrix},$$

¹We have considered that $\mu_1 \neq \mu_2$ when the relative distance between μ_1 and μ_2 is greater than 1% in order to cope with numerical accuracy problems.

N	Noise level [pixel]	Type	# cases $\mu_1 \neq \mu_2$	$\max(\xi)$	$mean(\mu_2)$ [pixel]	$std(\mu_2)$ [pixel]
10	1	P	0	0.000	1.620	0.489
10	1	L	0	0.000	1.542	0.487
10	1	E	0	0.000	1.482	0.486
10	2	P	0	0.000	1.801	0.401
10	2	L	0	0.000	1.665	0.396
10	2	E	0	0.000	1.552	0.391
10	3	P	0	0.000	2.220	0.396
10	3	L	4	0.015	2.044	0.388
10	3	E	0	0.000	1.877	0.400
30	1	P	0	0.000	1.841	0.404
30	1	L	0	0.000	1.749	0.403
30	1	E	5	0.024	1.681	0.404
30	2	P	1	0.000	2.030	0.336
30	2	L	0	0.000	1.879	0.340
30	2	E	0	0.000	1.748	0.341
30	3	P	0	0.000	2.383	0.351
30	3	L	0	0.000	2.177	0.350
30	3	E	0	0.000	1.988	0.342
50	1	P	12	0.020	1.887	0.328
50	1	L	3	0.015	1.799	0.326
50	1	E	17	0.055	1.735	0.327
50	2	P	6	0.025	2.080	0.304
50	2	L	0	0.000	1.937	0.304
50	2	E	0	0.000	1.810	0.303
50	3	P	0	0.000	2.464	0.330
50	3	L	0	0.000	2.273	0.326
50	3	E	0	0.000	2.091	0.316

Table 2: Results with 10800 random configurations obtained for different number of views ($N = 10, 30, 50$), level of noise ($\eta = 1, 2, 3$ pixels) and type of data (“P”: points; “L”: $N/2$ points and $N/2$ lines segments; “E”: $N/2$ points and $N/2$ discs). The table shows the number of views N , level of noise η , type of data, number of cases where $\mu_1 \neq \mu_2$, maximum value of the relative error ξ , mean of μ_2 , and standard deviation of μ_2 .

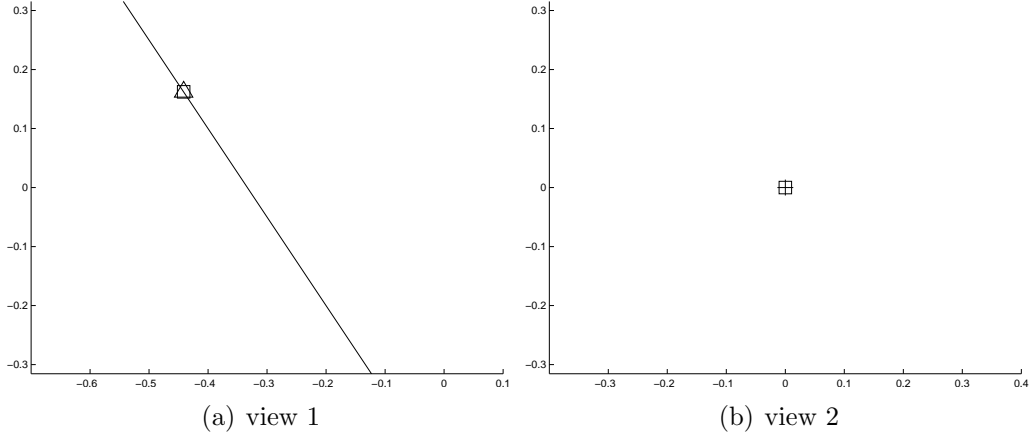


Figure 5: An example with multiple solutions: problem data (line in figure a and cross in figure b) and projections of $\hat{\mathbf{X}}$ (squares). The triangle is the closest point of the line to the projection of $\hat{\mathbf{X}}$.

it turns out that \mathcal{A}_1 is a portion of the epipolar line corresponding to $\hat{\mathbf{x}}_2$. Hence, the triangulation problem has multiple solutions in this case since any point of \mathcal{A}_1 is the projection on the first view of a 3D point whose projection on the second view is $\hat{\mathbf{x}}_2$.

We use the TFML method, and we obtain $\mu_1 = 0.000$. In this case, the minimum eigenvalue of the matrix $\bar{\mathbf{Q}}$ has multiplicity greater than 1, and hence the vector $\bar{\mathbf{w}}$ in (31) is not unique. In particular, there are 2 eigenvalues equal to $\lambda_{\min}(\bar{\mathbf{Q}})$, and hence $\bar{\mathbf{w}}$ is a free point on a line of the form

$$\bar{\mathbf{w}} = \bar{\mathbf{w}}_0 + a\bar{\mathbf{w}}_1$$

(since a degree of freedom is frozen by imposing that the last entry is 1). We simply select² $a = -1$, and we proceed with the so obtained $\bar{\mathbf{w}}$ that yields

$$\mu_1 = 0.000, \quad \mu_2 = 0.000, \quad \hat{\mathbf{X}} = (-0.613, -0.225, 0.388)^T.$$

This obtained 3D point is hence a solution of the considered triangulation problem. Figure 5 illustrates the found solution.

By selecting other values of a in the construction of $\bar{\mathbf{w}}$, the method returns other admissible 3D points. For instance, with $a = 1$ one finds

$$\mu_1 = 0.000, \quad \mu_2 = 0.000, \quad \hat{\mathbf{X}} = (4.194, -9.389, 5.194)^T.$$

²The admissible range for a is the set of values for which $\bar{\mathbf{x}}_1$ belongs to \mathcal{A}_1 .

4.5. Real Data: Dinosaur, Corridor, House, and College

Here we consider well-known examples with real data, in particular:

- the dinosaur example, which is a turntable sequence of 36 views of a toy dinosaur with 4983 points;
- the corridor example, which is a sequence of 11 views with 737 points;
- the house example, which is a sequence of 10 views with 672 points;
- the college example, which is a sequence of 5 views with 1331 points.

Each 3D point is estimated by solving the triangulation problem with all the available views. Tables 3–6 show the obtained results with the TFML method. As we can see, μ_1 is always optimal except for one point in the corridor example. Specifically, this point is visible in 3 views, and the method described in Section 3.1 yields

$$\mu_1 = 0.163, \quad \mu_2 = 0.175.$$

We hence use the TPML method, hence getting the exact solution described by

$$\mu_3 = 0.170, \quad \mu_4 = 0.170, \quad \hat{\mathbf{X}} = (-3.849, 4.718, -19.709)^T.$$

These tables also show the average computational time (ACT) in Matlab (on a standard personal computer with Intel Pentium 4 and Windows XP), which is indeed small especially for small number of views. Figures 6–9 show some pictures of the sequences and the obtained 3D points.

4.6. Real Data: Monna Lisa and Book

Lastly, we consider two situations with real data in the presence of occlusions. The first situation is shown in Figures 10a, 10c and 10e. The point we want to estimate via triangulation is the extreme of the right index finger of Monna Lisa, see the corresponding zoom-in areas in Figures 10b, 10d and 10f. This point is visible in Figures 10c and 10e, but is occluded in Figure 10a. Nevertheless, a region where this point obviously lies in can be easily identified, for instance via an ellipse. In order to show the behavior of the propose method with respect to the size of such a region, we consider the seven ellipses shown in Figure 10b.

Figures 11a, 11c and 11e show the second situation. The point we want to estimate via triangulation is the top-left corner of the letter “N”, see the



(a)



(b)

Figure 6: Dinosaur example: an image of the sequence and the reconstructed model by using for each 3D point all the available views.

N	# points	# cases $\mu_1 \neq \mu_2$	$mean(\mu_2)$ [pixel]	$std(\mu_2)$ [pixel]	ACT [s]
2	2300	0	0.121	0.104	0.010
3	1167	0	0.323	1.024	0.048
4	584	0	0.498	1.721	0.060
5	375	0	0.410	1.011	0.071
6	221	0	0.669	1.928	0.080
7	141	0	0.459	1.138	0.097
8	88	0	0.646	1.859	0.115
9	44	0	0.433	0.560	0.148
10	26	0	0.450	0.484	0.175
11	15	0	1.100	2.867	0.215
12	14	0	0.389	0.191	0.270
13	5	0	1.103	0.940	0.303
14	2	0	0.375	0.248	0.390
21	1	0	0.822	0.000	1.094

Table 3: Results for the dinosaur example: number of views N , number of 3D points with N views, number of cases where $\mu_1 \neq \mu_2$, mean of μ_2 , standard deviation of μ_2 , and ACT.



(a)



(b)

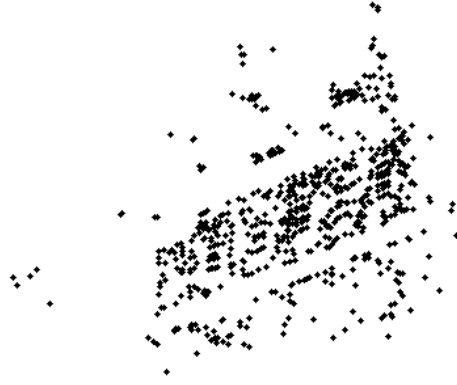
Figure 7: Corridor example: an image of the sequence and the reconstructed model by using for each 3D point all the available views.

N	# points	# cases $\mu_1 \neq \mu_2$	$mean(\mu_2)$ [pixel]	$std(\mu_2)$ [pixel]	ACT [s]
3	341	1	0.166	0.137	0.045
5	146	0	0.223	0.160	0.078
7	88	0	0.260	0.206	0.133
9	58	0	0.258	0.221	0.220
11	104	0	0.262	0.144	0.307

Table 4: Results for the corridor example: number of views N , number of 3D points with N views, number of cases where $\mu_1 \neq \mu_2$, mean of μ_2 , standard deviation of μ_2 , and ACT.



(a)

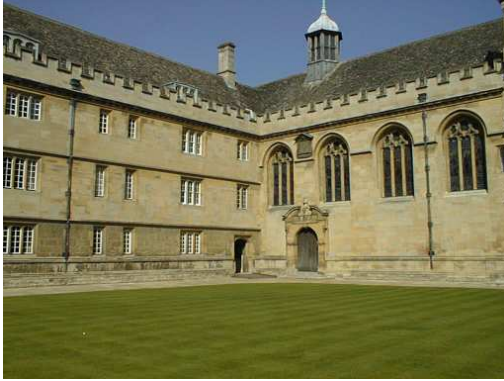


(b)

Figure 8: House example: an image of the sequence and the reconstructed model by using for each 3D point all the available views.

N	# points	# cases $\mu_1 \neq \mu_2$	$mean(\mu_2)$ [pixel]	$std(\mu_2)$ [pixel]	ACT [s]
3	382	0	0.209	0.143	0.056
4	19	0	0.372	0.125	0.083
5	158	0	0.294	0.506	0.073
6	3	0	0.625	0.237	0.089
7	90	0	0.374	0.320	0.109
8	1	0	0.686	0.000	0.172
9	12	0	0.476	0.270	0.185
10	7	0	0.266	0.104	0.230

Table 5: Results for the house example: number of views N , number of 3D points with N views, number of cases where $\mu_1 \neq \mu_2$, mean of μ_2 , standard deviation of μ_2 , and ACT.



(a)



(b)

Figure 9: College example: an image of the sequence and the reconstructed model by using for each 3D point all the available views.

N	# points	# cases $\mu_1 \neq \mu_2$	$mean(\mu_2)$ [pixel]	$std(\mu_2)$ [pixel]	ACT [s]
2	1052	0	0.098	0.077	0.009
3	215	0	0.143	0.063	0.041
4	50	0	0.157	0.055	0.063
5	14	0	0.165	0.030	0.080

Table 6: Results for the college example: number of views N , number of 3D points with N views, number of cases where $\mu_1 \neq \mu_2$, mean of μ_2 , standard deviation of μ_2 , and ACT.

corresponding zoom-in areas in Figures 11b, 11d and 11f. This point is visible in Figures 11c and 11e, but is occluded in Figure 11a. Nevertheless, a region where this point obviously lies in can be easily identified, for instance via a line segment. Analogously to the previous case, we consider several line segments with different length as shown in Figure 11b.

Tables 7 and 8 show the obtained results for these two examples. As can be seen, the image error decreases by increasing the size of the uncertain region (either the ellipse for Monna Lisa or the line segment for the book). This is expected since these regions are nested, i.e. the region for a given size is contained in the regions for larger sizes. Clearly, this does not imply that the accuracy of the estimated 3D point increases by increasing the size of the uncertain region since, beyond a certain size, only image points that obviously do not match the sought one are included. As a rule, hence, one should select the uncertain regions as small as possible compatibly with the available views in order to avoid to include meaningless points that may affect the optimal solution of the triangulation problem. This means that, for the present cases, the best uncertain regions to consider are probably the third smallest ellipse and the third shortest line segment.

5. Conclusion

This paper has proposed a new strategy to multiple-view L_2 triangulation, in the standard case where all image points are available, and in the case where some image points are not available. Two methods have been proposed based on LMIs, namely the TFML and the TPML methods, which exploit respectively the fundamental matrices and the projection matrices. In particular, the TFML method is fast (the average computational time



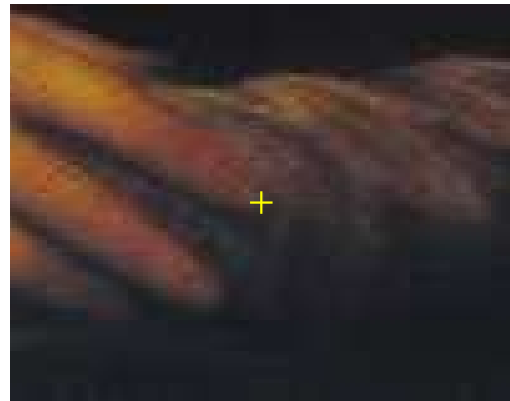
(a)



(b)



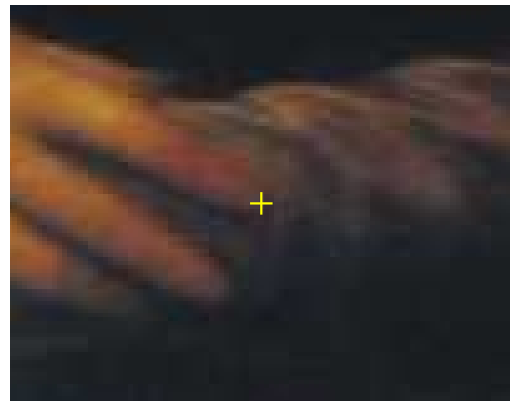
(c)



(d)



(e)



(f)

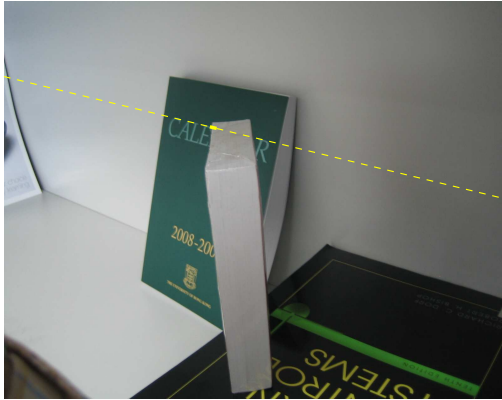
Figure 10: Monna Lisa example: the point of interest is visible in the second and third views but it is occluded in the first view where elliptical regions of various size are used for the triangulation problem.

Ellipse number	μ_1 [pixel]	μ_2 [pixel]	$\hat{\mathbf{X}}$
1	3.983	3.983	$(0.0769, -0.451, 2.739)^T$
2	3.247	3.247	$(0.0743, -0.448, 2.733)^T$
3	2.551	2.551	$(0.0722, -0.445, 2.727)^T$
4	1.930	1.930	$(0.0709, -0.443, 2.724)^T$
5	1.445	1.445	$(0.0718, -0.442, 2.726)^T$
6	1.157	1.158	$(0.0756, -0.443, 2.734)^T$
7	1.062	1.062	$(0.0810, -0.446, 2.747)^T$

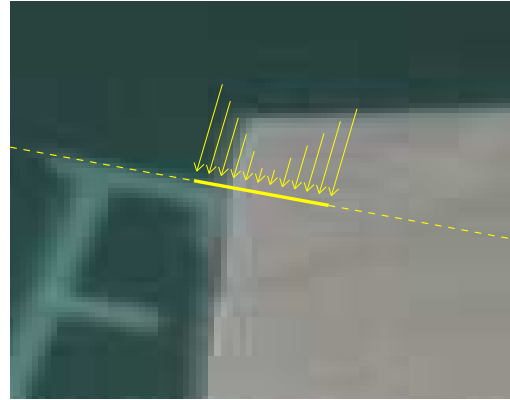
Table 7: Results for Monna Lisa example: ellipse number (1 is the smallest and 7 is the largest in Figure 10b), lower bound μ_1 , upper bound μ_2 , and candidate solution $\hat{\mathbf{X}}$. Since $\mu_1 = \mu_2$, one has that $\mu^* = \mu_1$, and $\hat{\mathbf{X}}$ is the solution of the triangulation problem.

Line segment number	μ_1 [pixel]	μ_2 [pixel]	$\hat{\mathbf{X}}$
1	3.265	3.265	$(-0.491, 0.867, 1.835)^T$
2	3.100	3.100	$(-0.491, 0.869, 1.837)^T$
3	2.632	2.632	$(-0.489, 0.871, 1.844)^T$
4	2.172	2.172	$(-0.488, 0.874, 1.850)^T$
5	1.725	1.725	$(-0.486, 0.876, 1.857)^T$
6	1.306	1.306	$(-0.485, 0.879, 1.864)^T$
7	0.951	0.951	$(-0.483, 0.881, 1.871)^T$

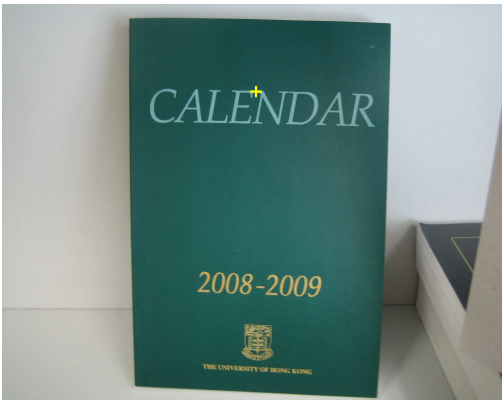
Table 8: Results for the book example: line segment number (1 is the shortest and 7 is the longest in Figure 11b), lower bound μ_1 , upper bound μ_2 , and candidate solution $\hat{\mathbf{X}}$. Since $\mu_1 = \mu_2$, one has that $\mu^* = \mu_1$, and $\hat{\mathbf{X}}$ is the solution of the triangulation problem.



(a)



(b)



(c)



(d)



(e)



(f)

Figure 11: Book example: the point of interest is visible in the second and third views but it is occluded in the first view where line segments of various length are used for the triangulation problem.

with two and three views is 0.01 and 0.05 seconds on Matlab) at the expense of possible conservatism that however occurs rarely in practice as shown by various examples with synthetic and real data. The Matlab code of the proposed approach will be soon available at <http://www.eee.hku.hk/~chesi>

Acknowledgement

The authors would like to thank the Editors and the Reviewers for their valuable comments. The work presented in this paper was supported in part by HKU Grant 200907176048 and RGC Grants HKU711208E and HKU712808E.

References

- [1] R. Hartley, P. Sturm, Triangulation, *Computer Vision and Image Understanding* 68 (2) (1997) 146–157.
- [2] R. Hartley, A. Zisserman, *Multiple view in computer vision*, Cambridge University Press, 2000.
- [3] H. Stewenius, F. Schaffalitzky, D. Nister, How hard is 3-view triangulation really?, in: *Int. Conf. on Computer Vision*, 2005, pp. 686–693.
- [4] M. Byrod, K. Josephson, K. Astrom, Fast optimal three view triangulation, in: *Asian Conf. on Computer Vision*, Tokyo, Japan, 2007.
- [5] F. Lu, R. Hartley, A fast optimal algorithm for l_2 triangulation, in: *Asian Conf. on Computer Vision*, Tokyo, Japan, 2007.
- [6] R. Hartley, Y. Seo, Verifying global minima for l_2 minimization problems, in: *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.
- [7] A. Bartoli, J.-T. Lapreste, Triangulation for points on lines, *Image and Vision Computing* 26 (2) (2008) 315–324.
- [8] R. Hartley, F. Schaffalitzky, l_∞ minimization in geometric reconstruction problems, in: *IEEE Conf. on Computer Vision and Pattern Recognition*, 2004, pp. 504–509.

- [9] Q. Ke, T. Kanade, Uncertainty models in quasiconvex optimization for geometric reconstruction, in: IEEE Conf. on Computer Vision and Pattern Recognition, 2006.
- [10] G. Chesi, A. Garulli, A. Tesi, A. Vicino, Homogeneous Polynomial Forms for Robustness Analysis of Uncertain Systems, Springer, 2009.
- [11] G. Chesi, LMI techniques for optimization over polynomials in control: a survey, IEEE Trans. on Automatic Control 55 (11) (2010) 2500–2510.
- [12] S. Boyd, L. El Ghaoui, E. Feron, V. Balakrishnan, Linear Matrix Inequalities in System and Control Theory, SIAM, 1994.
- [13] G. Stengle, A nullstellensatz and a positivstellensatz in semialgebraic geometry, Math. Ann. 207 (1974) 87–97.
- [14] G. Chesi, Y. S. Hung, Image noise induced errors in camera positioning, IEEE Trans. on Pattern Analysis and Machine Intelligence 29 (8) (2007) 1476–1480.
- [15] G. Chesi, Camera displacement via constrained minimization of the algebraic error, IEEE Trans. on Pattern Analysis and Machine Intelligence 31 (2) (2009) 370–375.